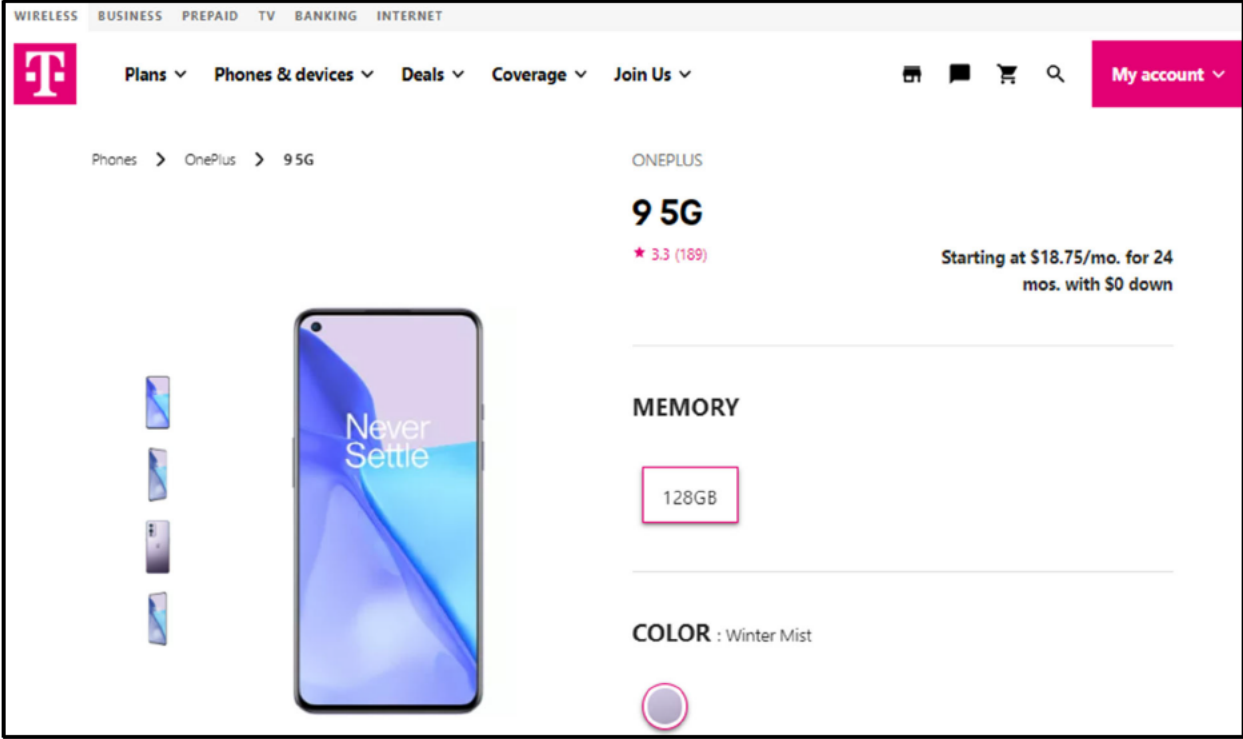


EXHIBIT 20

U.S. Patent No. 7,930,539

Claim 1	Identification
<p>1[pre] A computer-implemented method for use in a computer system including a plurality of resources, the method comprising steps of:</p>	<p>To the extent the preamble is limiting, the OnePlus 9 uses an ARMv8-A based processor that implements a method for use in a computer system including a plurality of resources comprising the steps below.</p>  <p>https://www.t-mobile.com/cell-phone/oneplus-9-5g</p>

OnePlus 9



Performance

Operating System: OxygenOS based on Android™ 11
CPU: Qualcomm® Snapdragon™ 888
5G Chipset: X60
GPU: Adreno 660
RAM: 8GB LPDDR5
Storage: 128GB UFS 3.1 2-LANE
Battery: 4,500 mAh (2S1P 2,250 mAh, non-removable)
Warp Charge 65T (10V/6.5A)
15W Wireless Charging

<https://www.oneplus.com/us/9/specs>

Snapdragon



888+ 5G mobile platform

The Snapdragon® 888+ 5G Mobile Platform fuels flagship experiences with profoundly intelligent entertainment, including AI-enhanced gameplay, streaming, photography, and more—backed by boosted performance, unrivaled speed and premium connectivity.



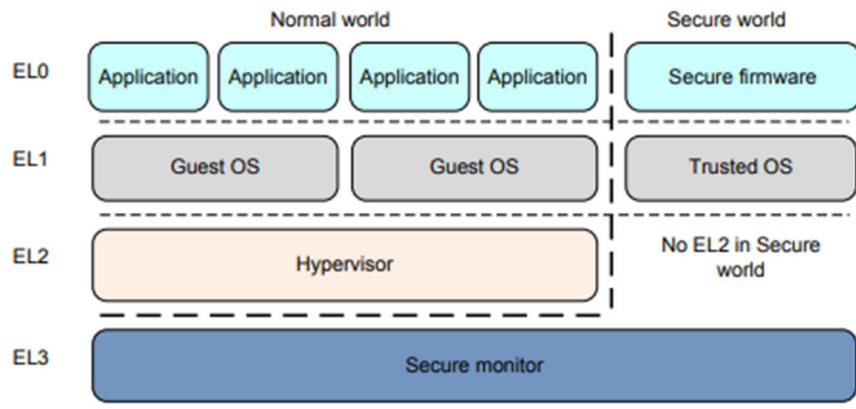
Unbridled performance for world-class entertainment

Better, faster, stronger—our newly upgraded Qualcomm® Kryo™ 680 CPU revs up your elite entertainment experiences. Open social feeds, load games, or access your favorite content in a flash with an industry-leading processor designed to meet high-end demands head-on. Plus, stay immersed for hours without stopping to recharge thanks to a hyper-efficient 5 nm process.

- The Qualcomm® Adreno™ 660 GPU renders detailed graphics in a flash
- Kryo 685 CPU with speeds up to 3.0 GHz* and Arm Cortex-X1-based architecture for increased efficiency

https://www.qualcomm.com/content/dam/qcomm-martech/dm-assets/documents/prod_brief_qcom_sd888_plus_5g_final.pdf

	<div data-bbox="596 191 1753 342"> <h2>About the core</h2> <p>The Cortex®-X1 core is a high-performance and low-power Arm product that implements the Arm®v8-A architecture.</p> </div> <p>https://developer.arm.com/documentation/101433/0102/Functional-description/Introduction/About-the-core</p> <div data-bbox="596 451 1169 500"> <h3>I Security in an ARMv8-A system</h3> </div> <p>A secure or trusted system is one that protects assets, for example passwords or credit card details from a range of plausible attacks, to prevent them from being copied or damaged, or made unavailable.</p> <p>Security is defined by the principles of:</p> <p>Security in an ARMv8-A System (https://developer.arm.com/documentation/100935/0100/Security-in-ARMv8-A-systems-?lang=en)</p>
[1] (A) receiving a request from a software program to access a specified one of the plurality of resources;	The system receives a request from a software program (such as a software application or OS) to access a resource, such as a normal world resource or a secure world resource.

	 <p style="text-align: center;">Figure 1 Security model for AArch64</p> <p>The ARM Architecture Reference Manual uses the terms Secure and Non-secure to refer to system security states. A Non-secure state does not automatically mean security vulnerability, but rather normal operation and is therefore the same as the Normal world. Typically, there is a master and slave relationship between Non-secure and Secure worlds. Code in the Secure world is only executed when the OS permits Secure world execution through a mechanism that is initiated by the Secure Monitor Call (SMC) instruction.</p>
<p>[1] (B) determining whether the specified one of the plurality of resources is a protected resource;</p>	<p>Security in an ARMv8-A System at 5</p> <p>A determination is made whether the specified one of the plurality of resources is a protected resource. For example, whether the resource is in secure world or normal world.</p> <div style="border: 1px solid black; padding: 10px; margin-top: 10px;"> <p>The ARM Security Extensions model allows system developers to partition device hardware and software resources, so that they exist in either the Secure world for the security subsystem, or the Normal world for everything else. Correct system design can ensure that no Secure world assets can be accessed from the Normal world. A Secure design places all sensitive resources in the Secure world, and ideally has robust software running that can protect assets against a wide range of possible software attacks.</p> </div> <p>Security in an ARMv8-A System at 5</p>
<p>[1] (C) if the specified one of the plurality of resources is a</p>	<p>If the specified one of the plurality of resources is a protected resource (e.g., in secure world), the steps below are performed.</p>

protected resource, performing steps of:	
<p>[1(C)] (1) if the computer system is operating in a protected mode of operation, then denying the request regardless of access rights associated with the software program including software programs having a most-privileged level; and</p>	<p>If the computer system is operating in a protected mode of operation, then the request is denied regardless of access rights associated with the software program including software programs having a most-privileged level. For example, if the Secure Monitor Disable bit is set to disable, Secure Monitor Call (SMC) instructions, required to access secure world resources, are disabled.</p> <p>D19.2.120 SCR_EL3, Secure Configuration Register</p> <p>The SCR_EL3 characteristics are:</p> <p>Purpose</p> <p>Defines the configuration of the current Security state. It specifies:</p> <ul style="list-style-type: none"> • The Security state of EL0, EL1, and EL2. The Security state is Secure, Non-secure, or Realm. • The Execution state at lower Exception levels. • Whether IRQ, FIQ,SError interrupts, and External abort exceptions are taken to EL3. • Whether various operations are trapped to EL3. <p>Arm Architecture Reference Manual for A-Profile Architecture at D19.2.29 (https://developer.arm.com/documentation/ddi0487/latest/)</p>

	<div>SMD, bit [7] Secure Monitor Call disable. Disables SMC instructions at EL1 and above, from any Security state and both Execution states, reported using an ESR_ELx.EC value of 0x00. 0b0 SMC instructions are enabled at EL3, EL2 and EL1. 0b1 SMC instructions are UNDEFINED. ———— Note ———— SMC instructions are always UNDEFINED at EL0. Any resulting exception is taken from the current Exception level to the current Exception level. If HCR_EL2.TSC or HCR.TSC traps attempted EL1 execution of SMC instructions to EL2, that trap has priority over this disable. ————— The reset behavior of this field is: <ul style="list-style-type: none">On a Warm reset, this field resets to an architecturally UNKNOWN value.</div> <div>Arm Architecture Reference Manual for A-Profile Architecture at D19-6959.</div> <div><div>The <i>Secure Monitor Call</i> (SMC) instruction provides software with a system call to EL3. When executing at a privileged Exception level, SMC instructions generates exceptions. For more information, see Secure Monitor Call (SMC) exception on page G1-9811 and SMC on page F5-8734.</div></div> <div><div>Figure G1-1 shows that when EL3 is using AArch32, the Exception levels and modes available in each Security state are as follows: <table><tr><th>Secure state</th><th></th></tr><tr><td>EL0</td><td>User mode.</td></tr><tr><td>EL3</td><td>Any mode that is available in Secure state, other than User mode.</td></tr></table></div></div> <div>Arm Architecture Reference Manual for A-Profile Architecture at G1-9748-49.</div>	Secure state		EL0	User mode.	EL3	Any mode that is available in Secure state, other than User mode.
Secure state							
EL0	User mode.						
EL3	Any mode that is available in Secure state, other than User mode.						
[1(C)] (2) processing the request based on the access	If the computer system is not operating in the protected mode of operation (e.g., SMD set to 0), then the request is processed based on access rights associated with the software program.						

rights associated with the software program if the computer system is not operating in the protected mode of operation.

The ARM Architecture Reference Manual uses the terms Secure and Non-secure to refer to system security states. A Non-secure state does not automatically mean security vulnerability, but rather normal operation and is therefore the same as the Normal world. Typically, there is a master and slave relationship between Non-secure and Secure worlds. Code in the Secure world is only executed when the OS permits Secure world execution through a mechanism that is initiated by the Secure Monitor Call (SMC) instruction.

Note

The use of the word world is used to describe not just the Execution state, but also all memory and peripherals that are accessible in that state. Non-secure memory and functions are also accessible to the Secure world.

The role of the Secure monitor is to provide a gatekeeper which manages the switches between the Secure and Non-secure worlds. In most designs its functionality is similar to a traditional operating system context switch, ensuring that state of the world that the core is leaving is safely saved, and the state of the world the processor is switching to is correctly restored.

Security in an ARMv8-A System at 5